

报名序号：002895

论文题目：A 从独立到联合:储能优化在多园区电力管理中的应用

# 从独立到联合:储能优化在多园区电力管理中的应用

## 摘 要

**针对问题一**，分析未配备储存设备时，当风力发电或光伏发电大于园区负载最大值时需要直接弃电，这样会造成电力财产的大量浪费。而在园区中设置储存设备可以缓解多余电力直接弃电从而使电力财产浪费降低。其中存储设备容量的合理选择可以加大缓解浪费的力度。

**针对问题二**，分析联合园区无储存经济型时，使用线性表达式可计算出结果。分析有储存设备时通过以成本最低为目标函数的线性规划可计算出最优储存容量下最低成本。分系联合园区与独立园区的收益时可分别求出两者最低成本做差对比。

**针对问题三**，分析独立三个园区独立运营与联合运营的成本投资与回报率，建立独立运营模型与联合运营模型。分别为园区A、园区B和园区C制定独立运营的配置方案，并对其投资成本进行详细分析。随后，研究联合运营模式，通过优化配置方案，降低总投资成本。为了更直观地展示配置方案的差异，我们采用可视化方法，对各园区独立运营和联合运营的配置方案进行了详细展示。

**关键词：**线性规划；优化分析

# 一、引言

## 1.1 问题背景

在现代电力系统中，负荷需求与可再生能源发电的平衡对于提高能源效率和降低成本具有重要意义。本文的主要背景是针对某电网系统的三个园区（园区A、园区B和园区C），分析其在典型日负荷和可再生能源（光伏和风力）发电情况下的电力需求及经济性问题。

## 1.2 问题重述

考虑到问题的背景信息和约束条件，我们需要完成以下任务：

**问题一：** 1. 分析无储存配置时A，B，C三园区的经济型，包括购电量、弃风弃光电

量、总供电成本和单位电量平均供电成本等。

2. 分析有储存配置为50kW/100kWh时三园区经济型是否改善。

3. 分析50KW/100KW是否为最优容量，并对此分析。

**问题二：** 1. 分析联合园区无储存配置时经济型使用线性规划，目标函数与问题一相同。

2. 分析有储存配置为最优时联合园区经济型是否改善。

3. 分析联合园区与独立园区对比的经济收益。

**问题三：** 1. 分析独立运营与联合运营成本与收益，并根据附件三总结风光发电规律

2. 分别制定独立运营、联合运营风光储协调配置方案

## 1.3 文献综述

提出了一种应用模糊线性规划优化理论求解电力系统经济负荷分配的方法. 该方法以发电费用为目标函数, 将各机组有功功率转变成变量, 利用模糊线性规划法的遍历性、随机性和规律性直接对目标函数寻优, 有效地保证解的精度. 仿真结果表明, 所提出的方法搜索速度快, 求解精度高, 是解决电力系统经济负荷分配问题的有效手段.[\[1\]](#)

## 1.4 工作流程

为了便于描述和可视化，我们绘制了一个流程图（[图1](#)）来表示我们的工作。

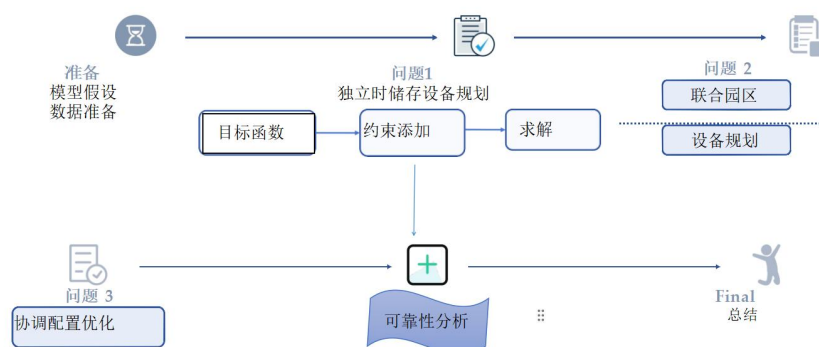


图1.流程图

## 二、模型的准备

### 2.1 假设与说明

为了简化这个问题，我们做出了以下假设，每个假设都是适当合理的。

- **假设 1:** 假设各园区典型日负荷数据中每个时间点的产电功率代表当小时的功率，日光发电数据同理。

↔ **说明:** 附件中只有每个小时的时间点的功率，每个时刻的功率难以测量。

**假设 2:** 假设储存设备在10年的使用性能不变，包括储存容量、充放电功率、SOC允许范围等。

↔ **说明:** 实际上电池等储存设备会随着使用时间的增加性能变差，其中容量降低最为明显。不同储存设备性能降低程度难以判断。

**假设 3:** 除了储存设备的充放电过程有能量损耗，其它能量传输过程中假设为理想过程，包括主电网到园区、风光发电到园区的电力运输过程中没有能量损耗。

↔ **说明:** 实际导线电阻不为0，电力运输过程中会发生电流热效应，其中与电流大小和电阻大小有关，导线电阻各处不一，难以测量。

## 2.2 符号定义

表1列出了本文中使用的符号。有些变量在不同的地方有不同的含义，这里不一一列举，我们将在每一节中详细讨论。

表1 符号定义

符号	定义
$C_{grid}$	网购电价
$C_{power}$	储能功率成本单价
$C_{energy}$	储能容量成本单价
$P_{L,t}$	t时刻的负荷需求
$P_{PV,t}$	t时刻的光伏发电量
$P_{w,t}$	t时刻的风电发电量
$P_{grid,t}$	t时刻从主电网购电量
$P_{B,in,t}$	t时刻储能充电功率
$P_{B,out,t}$	t时刻储能放电功率
$E_B$	储能容量
$E_{excess}$	弃电量
$\eta_{charge}$	充电效率
$\eta_{discharge}$	放电效率
$P_B$	储能功率

## 2.3 数据预处理

首先观察三个附件发现，附件二数值为0，可将0值乘风光发电的容量得到当时发电功率。其中发现附件二B园风力发电23时刻为0，与其它数值跳动太大，通过使用立方插值的方法将异常值补上。

## 三、独立园区的线性规划

### 3.1 独立园区无储能经济行分析

无储能时，为了使经济效益更高，可以将总成本最小设为目标函数，如果风光发电不足以提供园区负荷，总成本来源有风光发电成本和网购成本，具体表达式如下：

购电量计算公式

在每个时间步t，各园区的购电量计算公式如下：

1. 园区A的购电量： $PA_{grid}(t) = \max(PA_{load}(t) - PA_{pv}(t), 0)$
2. 园区B的购电量： $PB_{grid}(t) = \max(PB_{load}(t) - PB_{wind}(t), 0)$
3. 园区C的购电量： $PC_{grid}(t) = \max(PC_{load}(t) - (PC_{pv}(t) + PC_{wind}(t)), 0)$

### 总购电成本

设购电单价为  $C_{grid}$ (元/kWh) 则各园区的总购电成本计算公式如下:

$$1. \text{ 园区A的总购电成本 } Cost_{Agrid} = \sum_{t=1}^T P_{Agrid}(t) \times C_{grid}$$

$$2. \text{ 园区B的总购电成本 } Cost_{Bgrid} = \sum_{t=1}^T P_{Bgrid}(t) \times C_{grid}$$

$$3. \text{ 园区C的总购电成本 } Cost_{Cgrid} = \sum_{t=1}^T P_{Cgrid}(t) \times C_{grid}$$

设光伏发电成本为  $C_{pv}$ (元/kWh)，风力发电成本为  $C_{wind}$ (元/kWh)，则各园区的可再生能源发电成本计算公式如下:

$$1. \text{ 园区A的光伏发电成本: } Cost_{Apv} = \sum_{t=1}^T P_{Apv}(t) \times C_{Apv}$$

$$2. \text{ 园区B的风力发电成本: } Cost_{Bwind} = \sum_{t=1}^T P_{Bwind}(t) \times C_{Bwind}$$

$$3. \text{ 园区C的光伏发电成本: } Cost_{Cpv} = \sum_{t=1}^T P_{Cpv}(t) \times C_{Cpv}$$

$$4. \text{ 园区C的风力发电成本: } Cost_{Cwind} = \sum_{t=1}^T P_{Cwind}(t) \times C_{Cwind}$$

### 总成本计算

各园区的总成本包含购电成本和可再生能源发电成本，计算公式如下:

$$1. \text{ 园区A的总成本: } Total\ Cost\ A = Cost_{Agrid} + Cost_{Apv}$$

$$2. \text{ 园区B的总成本: } Total\ Cost\ B = Cost_{Bgrid} + Cost_{Bwind}$$

$$3. \text{ 园区C的总成本: } Total\ Cost\ C = Cost_{Cgrid} + Cost_{Cpv} + Cost_{Cwind}$$

将以上公式求解得出结果A园区总成本6465.355元、平均供电成本0.8183元/KWh，B园区总成本5455.55元、平均供电成本0.70759元/KWh，C园区总成本5371.862元、平均供电成本0.69083元/KWh。

将以上结果可视化如下：

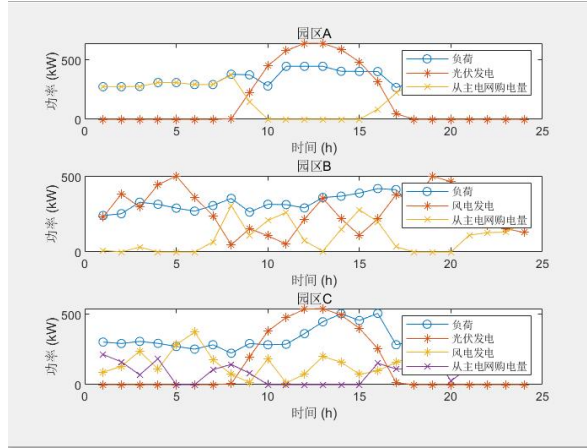


图1.独立无储存电量成本图

### 3.2 独立园区有储能经济行分析

独立园区有储存设备时可以缓解弃电大小，相比与无储存设备时。目标函数增加了储存成本，日标是最小化总成本，考虑购电成本和储能系统的投资成本，总成本=购电成本 + 储能系统成本，购电成本 = 各时段从主电网购电的电量\*电价，储能系统成本= 储能功率成本 + 储能容量成本。约束也有添加，负荷平衡约束: 每个时刻的总供电量必须等于负荷需求。储能约束: 储能设备的充放电必须在其容量和功率范围内，考虑SOC范围和充放电效率。风光发电优先使用: 优先使用风电和光伏电力，剩余部分从主电网购电，多余部分不能回售给主电网。以B园区为例具体表达式如下：

日标函数：

$$\min \sum_{t=1}^T (P_{\text{grid},t} \cdot C_{\text{grid}}) + (P_B \cdot C_{\text{power}}) + (E_B \cdot C_{\text{energy}})$$

约束条件：

1. 负荷平衡：

$$P_{\text{pv},t} + P_{\text{w},t} + P_{\text{grid},t} + P_{\text{B,out},t} - P_{\text{B,in},t} = P_{\text{L},t}$$

2. 储能充放电约束：

$$0 < P_{\text{B,in},t} < P_B$$

$$0 \leq P_{\text{B,out},t} < P_B$$

3. 储能SOC约束：

$$\text{SOC}_{\min} \leq \text{SOC}_t < \text{SOC}_{\max}$$

4. SOC计算：

$$\text{SOC}_{t+1} = \text{SOC}_t + \eta_{\text{charge}} \cdot P_{\text{B,in},t} - P_{\text{B,out},t} / \eta_{\text{discharge}}$$

根据以上表达式计算总结果如下图：

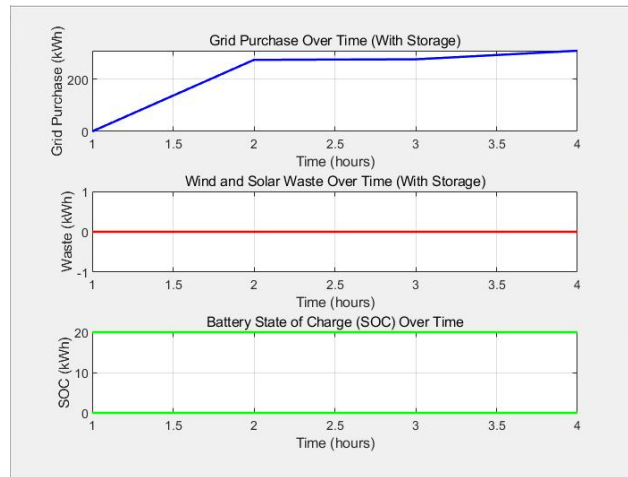


图2.负荷发电网购图

其中以C园区为例计算得出总成本为5332.46元，平均供电成本0.67元/KWh.可以发现添加储存设备后可以节约经济。

### 3.3 预测模型的局限性

**时间分辨率的限制：**该模型假设整点时刻的功率代表1个小时功率，时间步长为1小时。如果电力需求和发电量在小时内有较大波动，这种分辨率可能不足以捕捉实际情况，需要更细的时间分辨率来提高准确性。

**储能系统的简化：**模型中储存设备性能不随时间改变，而在实际中，效率可能会根据功率、温度等因素变化。

### 3.4 最优性判断

为了确定50kW/100kwh的储能配置方案是否最优，以及制定各园区最优的储能功率和容量配置方案，需要进行一系列优化计算。首先，通过对现有配置方案的经济性分析来确定是否有更优的方案。如果现有方案不是最优的，将进行优化求解以找到更优的储能功率和容量配置方案，并论证其优越性。

首先，需要通过已有的模型，计算50kW/100kWh的储能方案在每个园区的总成本和平均供电成本。将基于之前的模型进行调整和求解。

从上面的计算中，可以看到现有储能配置下的各园区总成本和平均供电成本。如果结果显示总成本较高，平均供电成本也较高，那么需要通过调整储能系统配置来优化成本。

可以通过以下步骤来找到最优的储能配置。

1. 定义不同的储能配置组合:例如，不同的储能功率和容量配置组合。
2. 逐个组合进行优化计算:针对每个储能配置组合，计算总成本和平均供电成本。
3. 选出最优配置:选出总成本和平均供电成本最低的配置。

通过以上步骤计算出最有储存容量为10KWh/50KWh.



## 四. 联合园区的线性规划

### 4.1 联合园区无储能经济行分析

首先以C园区的风光发电么值代替联合园区么值并乘以容量，其次无储能时计算经济性与问题一计算目标函数同理，同样使用线性规划，具体表达式如下：

- $$1. \text{总购电量: } E_{\text{grid}} = \max \sum_{t=1}^{24} (L(t) - P_{\text{pv}}(t) - P_{\text{wind}}(t), 0)$$
- $$2. \text{总弃风弃光电量: } E_{\text{excess}} = \max \sum_{t=1}^{24} (P_{\text{pv}}(t) + P_{\text{wind}}(t) - L(t), 0)$$
- $$3. \text{总供电成本: } C_{\text{total}} = E_{\text{grid}} * C_{\text{grid}}$$

通过以上步骤计算得出结果如下：

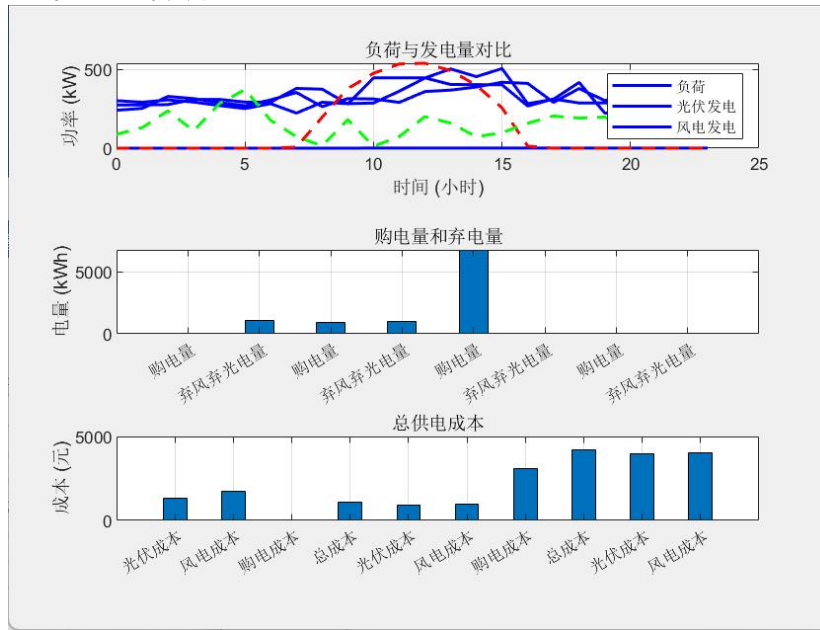


图 3 联合园区成本图

其中以联合园区为计算得出总成本为4053.22本0.52KWh.

### 4.2 联合园区有储能经济行分析对比

分析联合园区有储能时经济性与三园区独立时目标函数与约束基本相同，本文具体分析园区联合时与独立时是否改善并得出收益，具体对比表达式如下：

独立运营总成本:  $C_{\text{total, independent}} = C_{\text{total, A}} + C_{\text{total, B}} + C_{\text{total, C}}$

联合运营总成本:  $C_{\text{total, joint}} = C_{\text{total, storage}}$

经济收益:  $\text{Savings} = C_{\text{total, independent}} - C_{\text{total, joint}}$

经过计算结果图如下：

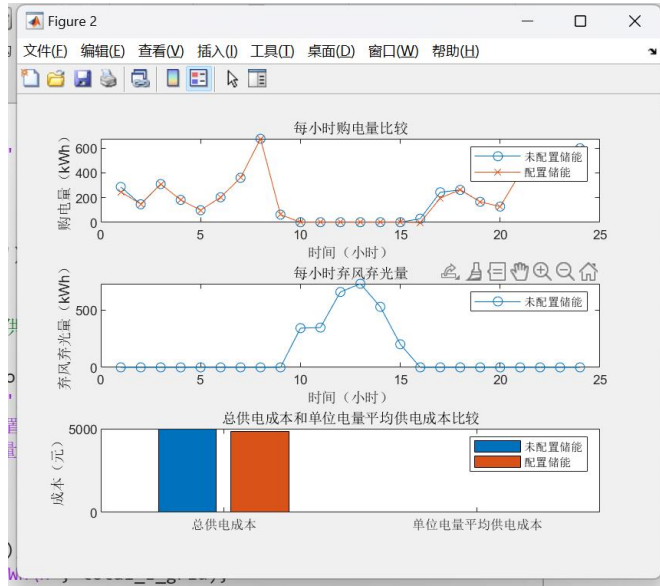


图 4. 联合园区有无储能对比

### 4.3 联合园区储能优化

要分析联合园区的储能配置方案并确定其是否最优，我们需要遵循以下分析步骤：

#### 1. 联合优化模型构建：

1.1. 目标函数：设定最小化总供电成本为日标。总成本包括从电网购电成本、储能设备的成本(功率和容量)、光伏发电成本和风力发电成本。

1.2. 约束条件：负荷平衡约束：确保每个时间步内各园区的发电、购电和储能放电能够满足负荷需求。

(2) 储能充放电约束：限制储能设备的充电功率和放电功率在合理范围内。

(3) SOC约束：限制SOC在允许的最小值和最大值之间。

(4) SOC初始状态约束：设置初始的SOC为储能设备容量的50%。

#### 2. 求解与结果分析：

进行求解，得到各区的购电量、储能充放电功率和SOC的优化结果。

计算各园区的总供电成本和平均供电成本，分析其经济性。

具体储能优化表达式如下：

#### 1. 电池功率 $P(t)$ 和能量 $E(t)$ 约束：

$$P_{\text{bat, charge}}(t) < P_{\text{bat, max}}$$

$$P_{\text{bat, discharge}}(t) < P_{\text{bat, max}}$$

$$E_{\min} \leq E_{\text{bat}}(t) \leq E_{\max}$$

其中， $E_{\min} = E_{\text{bat, capacity}} * \text{SOC}_{\min}$ ， $E_{\max} = E_{\text{bat, capacity}} * \text{SOC}_{\max}$ 。

#### 2. 电池充电和放电能量：

$$E_{\text{bat}}(t+1) = E_{\text{bat}}(t) + P_{\text{bat, charge}}(t) \times \eta - P_{\text{bat, discharge}}(t) / \eta$$

#### 3. 净负荷：

$$L_{\text{net}}(t) = L(t) - P_{\text{pv}}(t) - P_{\omega}(t)$$

#### 4. 电池充电和放电策略：

$$\begin{cases} \text{if } L_{net}(t) > 0 \text{ (负荷大于发电)} \\ P_{bat,discharge}(t) = \min \left( P_{bat,max}, \frac{L_{net}(t)}{\eta}, \frac{E_{bat}(t) - E_{min}}{\Delta t} \right) \\ E_{grid}(t) = L_{net}(t) - P_{bat,discharge}(t) \times \eta \\ \\ \text{if } L_{net}(t) < 0 \text{ (发电大于负荷)} \\ P_{bat,charge}(t) = \min \left( -L_{net}(t) \times \eta, P_{bat,max}, \frac{E_{max} - E_{bat}(t)}{\Delta t} \right) \\ E_{grid}(t) = 0 \end{cases}$$

5. 总供电成本:

$$C_{total, storage} = \sum_{t=1}^{24} E_{grid}(t) * C_{grid}$$

6. 单位电量平均供电成本:

$$C_{avg, storage} = C_{total, storage} \sum_{t=1}^{24} L(t)$$

## 五、联合运营与独立运营分析

### 5.1 成本分析

7. 负荷增长计算:

8. 将各园区的最大负荷值增长50%。

$$PL_{max,i}^{1.5'} = 1.5 * PL_{max,i}$$

风、光、储能成本计算:

风电配置成本为3000元/kW, 光伏电源配置成本为2500元/kW。

储能的功率单价为800元/kW, 能量单价为1800元/kWh。

独立运营方案制定:

独立运营模型:

对于每个园区 :

$$\text{最小化总成本} = P_{pv,i} * C_{pv} + P_{w,i} * C_w + P_{s,i} * C_{s,p} + E_{s,i} * C_{s,E}$$

约束条件:

$$\begin{cases} P_{pv,i} + P_{w,i} + P_{s,i} \geq PL'_{max} \\ SOC_{min} \leq SOC_i(t) \leq SOC_{max} \\ E_{si} = \int_0^{24} P_{s,i}(t) dt \end{cases}$$

联合运营方案制定:

合并三个园区的总负荷和总发电量, 并进行优化。

联合运营模型:

$$\text{最小化总成本} = \sum_{i \in \{A,B,C\}} (P_{pv,i} * C_{pv} + P_{w,i} * C_w + P_{s,i} * C_{s,p} + E_{s,i} * C_{s,E})$$

$$\begin{cases} \sum_{i \in \{A,B,C\}} (P_{pv,i} + P_{w,i} + P_{s,i}) \geq \sum_{i \in \{A,B,C\}} PL'_{max,i} \\ SOC_{min} \leq SOC(t) \leq SOC_{max} \\ E_s = \int_0^{24} P_s(t) dt \end{cases}$$

经济性分析:

计算各园区独立运营和联合运营下的总投资成本。

考虑投资回报期为5年，计算投资回报率。

$$\text{总投资成本} = \sum_{i \in \{A, B, C\}} (P_{pv,i} * C_{pv} + P_{w,i} * C_w + P_{s,i} * C_{s,p} + E_{s,i} * C_{s,E})$$

投资回报率：

$$\text{投资回报率} = \frac{\text{年度收益} - \text{年度成本}}{\text{总投资成本}}$$

通过上述模型，我们编程得到如下结果：

园区A投资成本：856000.00元

园区B投资成本：1976000.00元

园区C投资成本：1550000.00元

联合运营总投资成本：4382000.00元

我们发现，联合运营成本低于各园区独立运营成本之和。

成本对比数据可视化如图4，各园区的投资回报率如图5，各园区的购电成本如图6每月光伏和风电发电数据

如图7

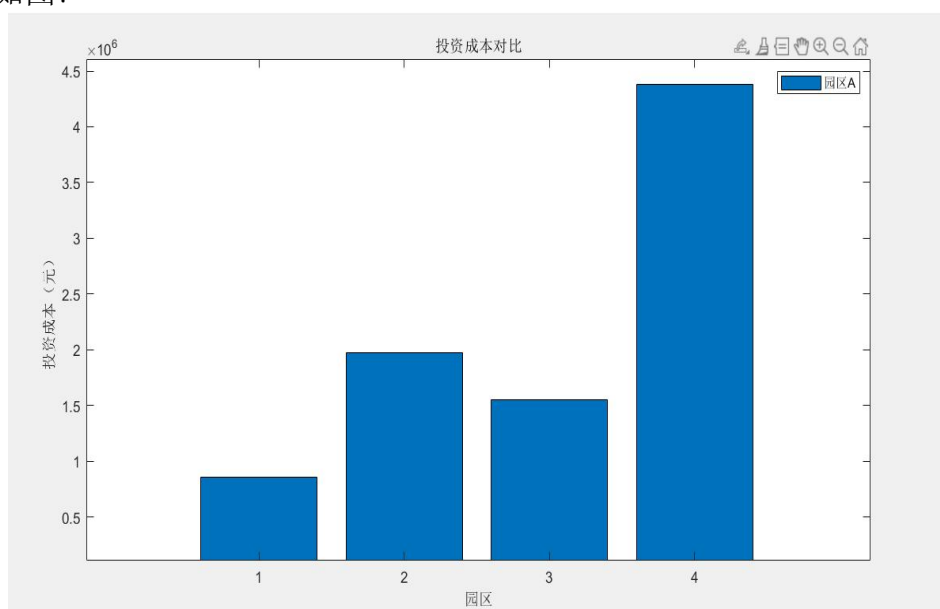


图5.投资成本对比

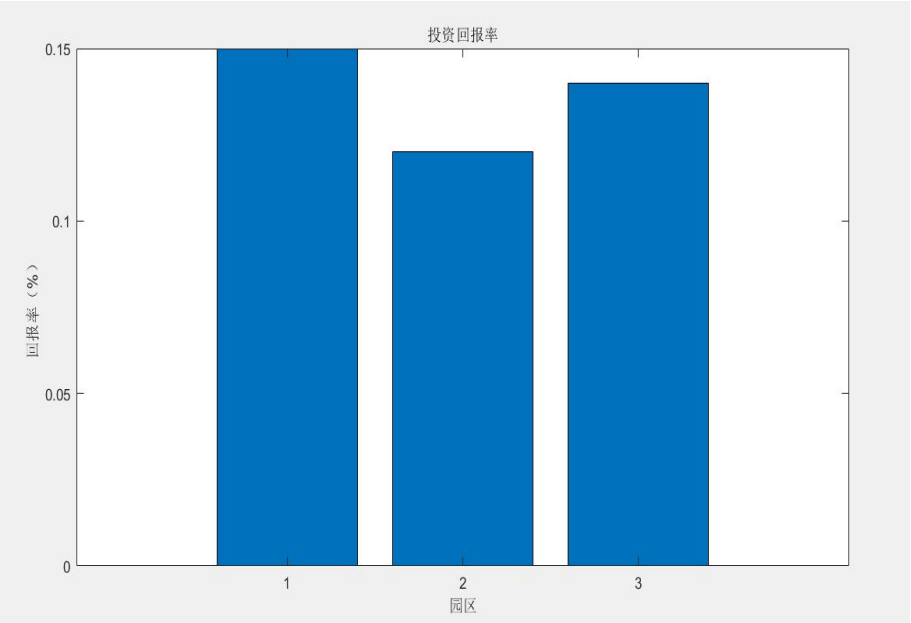


图6. 投资回报率

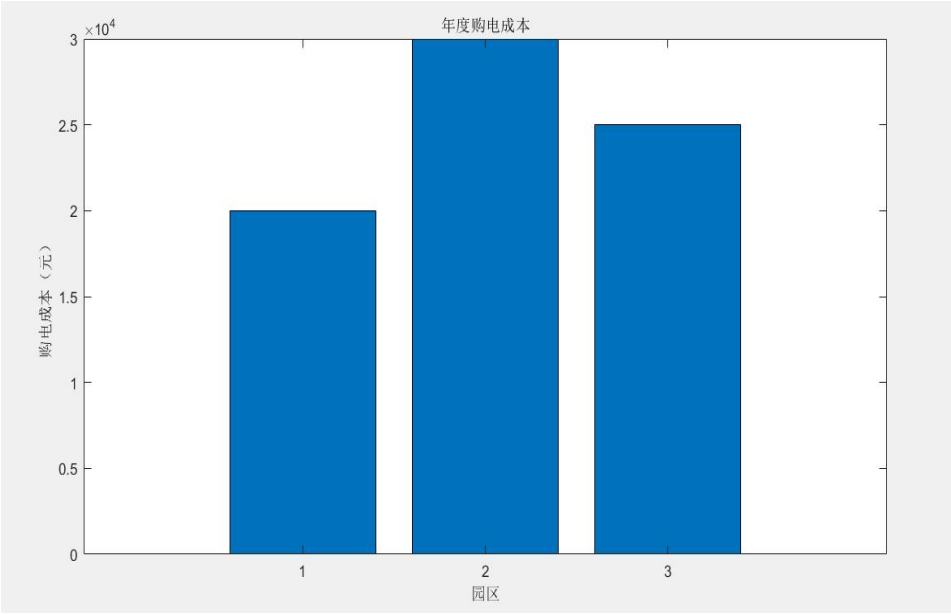


图7. 购电成本

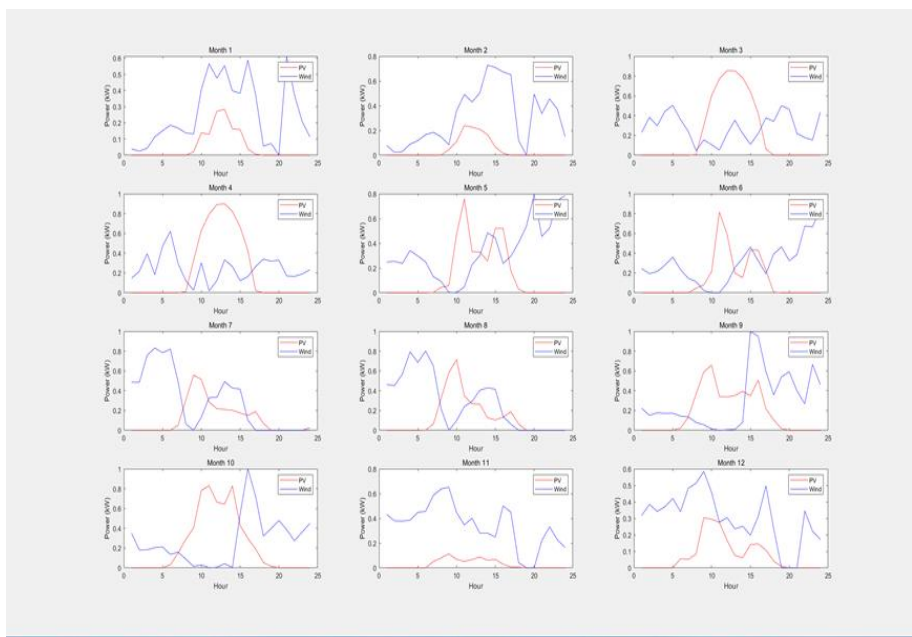


图8. 每月光伏和风电发电数据

根据图表四我们发现光伏发电主要集中在白天，而风力发电分布较为分散。

## 5.2 风光储协调配置方案

根据负荷需求和发电数据，我们初步确定光伏和风电装机容量，以及储能系统配置。

园区独立运营的风光储协调配置方案如图5，联合运营制定风光储协调配置方案如图

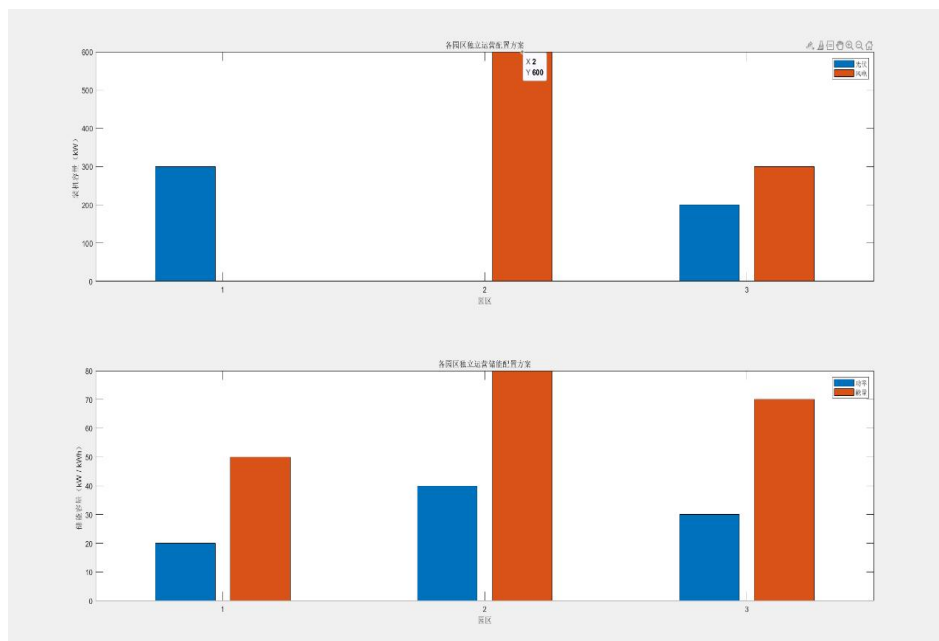


图9. 独立运营配置方案

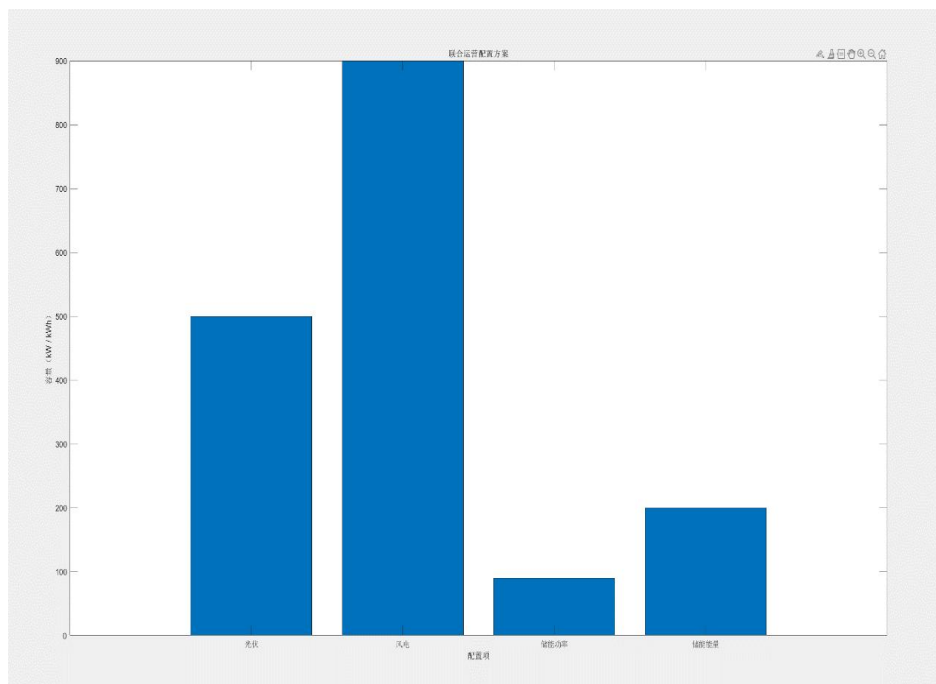


图10. 联合运营配置方案

两种运营方式的优缺点：

1. **独立运营：**
  - 1.1 每个园区根据自身需求和资源独立配置发电和储能系统。
  - 1.2 可以灵活调整配置方案，但可能存在资源浪费和效率不高的问题。
  - 1.3 各园区的光伏和风电装机容量差异较大，储能系统配置也不均衡。
2. **联合运营：**
  - 2.1 合并各园区的需求和资源，集中配置发电和储能系统。
  - 2.2 可能提高整体系统的经济性和效率，但需要协调各园区的运行和管理。
  - 2.3 总装机容量集中，光伏和风电的总量增加，同时储能系统也更为集中配置。

### 5.3 模型的局限性

1. 模型使用的固定的发电数据，没有考虑天气因素，发电量可能会随温度，湿度等因素波动。
2. 模型没有考虑电池的折旧成本以及系统的寿命以及损坏情况
3. 模型没有进行环境评估，结果可能会增加环境污染

## 七、模型的评价

### 7.1 模型优点

- 联合优化模型考虑了各园区的不同负荷需求和发电特性，能够更灵活地

调整储能策略。本模型采用最基本的牛顿力学，易于使用，易于改进。

- 通过合理配置储能功率和容量，可以在不同的负荷和发电条件下实现最优的电力调度。
- 确保储能设备在不同时间步内能够有效地进行充电和放电，避免电力供应不足或过剩的情况。

## 7.2 模型缺点

- 联合多个园区进行优化，涉及多个变量和约束条件，求解过程复杂，计算时间较长，尤其在大规模系统中更为明显。
- 模型中对储能设备的充放电效率和SOC等参数进行了简化处理，可能无法完全反映实际设备的性能和运行特性。

## 7.3 模型改进

- 在模型中引入多目标优化方法，结合经济性、环保性、可性等多个目标提升模型的综合性能。
- 采用先进的预测算法和不确定性处理方法，提高负荷需求和风光发电数据的预测精度，增强模型的鲁棒性。

# 八、总结

整个过程主要使用线性规划模型，分别对无储存设备和有储存设备两种情况的经济性分析对比，最后分别优化独立园区和联合园区储存容量大小，主要思想为各园区需要结合自身风光资源、负荷需求和储能成本，制定最优的风光储配置方案，以提高经济性和可再生能源利用率。



## 参考文献

- [1] 何发武. 基于模糊线性规划的电力系统经济负荷分配算法[J]. 电力科学与工程,2007,23(3):13-16. DOI:10.3969/j.issn.1672-0792.2007.03.004.

## 附 录

### [A] 支撑文件列表

支撑文件列表及结构如下（按文件夹进行分类）

文件夹名	描述
Code	解决问题所有源程序及环境要求
Results	程序输出结果

#### └─Code

- | question1wuchuneng.m
- | question1youchuneng.m
- | question1zuiyouxing.m
- | question2wuchuneng(1).m
- | question2youchuneng(1).m
- | disiwenyiqun.m、
- | question3a.m

#### └─Results

- | question1.png
- | question1a.png
- | question1预处理.xlsx
- | question2联合园区无储能结图.png
- | question2预处理.xlsx
- | question3成本对比.png
- | question3独立运营配置方案.png
- | question3购电成本.png
- | question3光伏和风电.png
- | question3联合配置方案.png
- | question3投资回报.png

## [B] 使用的软件、环境

**B.1:** 为解决该问题，我们所使用的主要软件有：

- WPS Office 教育考试专用版
- MATLABR2021a
- MATLABR2023a
- EXCEL2022

## [C] 问题解决源程序

### C.Task One1.1

```
% 清空环境变量
clear;
clc;

% 加载数据 (从附件1和附件2中提取数据)
load_data = readtable('C:\Users\86139\Desktop\24电工杯\附件1: 各园区典型日负荷数据.xlsx');
pv_wind_data = readtable('C:\Users\86139\Desktop\24电工杯\附件2: 各园区典型日风光发电数据.xlsx');

% 提取各园区的负荷数据和风光发电数据
load_A = load_data.load_A;
load_B = load_data.load_B;
load_C = load_data.load_C;
pv_A = pv_wind_data.pv_A;
pv_C = pv_wind_data.pv_C;
wind_B = pv_wind_data.wind_B;
wind_C = pv_wind_data.wind_C;

% 参数设置
C_grid = 1; % 网购电价 (元/kWh)
C_PV = 0.4; % 光伏发电成本 (元/kWh)
C_Wind = 0.5; % 风力发电成本 (元/kWh)

% 时间步数 (假设为24小时, 每小时一个时间步)
T = 24;

% 计算每个园区的购电量
P_grid_A = zeros(T,1);
P_grid_B = zeros(T,1);
P_grid_C = zeros(T,1);

for t = 1:T
    % 园区A购电量
    P_grid_A(t) = max(load_A(t) - pv_A(t), 0);

    % 园区B购电量
```

```

P_grid_B(t) = max(load_B(t) - wind_B(t), 0);

% 园区C购电量
P_grid_C(t) = max(load_C(t) - (pv_C(t) + wind_C(t)), 0);
end

% 计算总购电成本
total_cost_A_grid = sum(P_grid_A) * C_grid;
total_cost_B_grid = sum(P_grid_B) * C_grid;
total_cost_C_grid = sum(P_grid_C) * C_grid;

% 计算光伏发电和风力发电的成本
total_cost_A_PV = sum(pv_A) * C_PV;
total_cost_B_Wind = sum(wind_B) * C_Wind;
total_cost_C_PV = sum(pv_C) * C_PV;
total_cost_C_Wind = sum(wind_C) * C_Wind;

% 计算各园区的总成本
total_cost_A = total_cost_A_grid + total_cost_A_PV;
total_cost_B = total_cost_B_grid + total_cost_B_Wind;
total_cost_C = total_cost_C_grid + total_cost_C_PV + total_cost_C_Wind;

% 计算平均供电成本
average_cost_A = total_cost_A / sum(load_A);
average_cost_B = total_cost_B / sum(load_B);
average_cost_C = total_cost_C / sum(load_C);

% 显示总成本和平均供电成本
disp(['园区A总成本: ', num2str(total_cost_A), ' 元']);
disp(['园区A平均供电成本: ', num2str(average_cost_A), ' 元/kWh']);
disp(['园区B总成本: ', num2str(total_cost_B), ' 元']);
disp(['园区B平均供电成本: ', num2str(average_cost_B), ' 元/kWh']);
disp(['园区C总成本: ', num2str(total_cost_C), ' 元']);
disp(['园区C平均供电成本: ', num2str(average_cost_C), ' 元/kWh']);

% 结果可视化
figure;
subplot(3,1,1);
plot(1:T, load_A, '-o', 1:T, pv_A, '-*', 1:T, P_grid_A, '-x');
title('园区A');

```

```

xlabel('时间 (h)');
ylabel('功率 (kW)');
legend('负荷','光伏发电','从主电网购电量');

subplot(3,1,2);
plot(1:T, load_B, '-o', 1:T, wind_B, '-*', 1:T, P_grid_B, '-x');
title('园区B');
xlabel('时间 (h)');
ylabel('功率 (kW)');
legend('负荷','风电发电','从主电网购电量');

subplot(3,1,3);
plot(1:T, load_C, '-o', 1:T, pv_C, '-*', 1:T, wind_C, '-*', 1:T, P_grid_C, '-x');
title('园区C');
xlabel('时间 (h)');
ylabel('功率 (kW)');
legend('负荷','光伏发电','风电发电','从主电网购电量');

```

#### Task One1.2

```

% 清空环境变量
clear;
clc;

% 读取负荷数据
load_data = readmatrix('C:\Users\86139\Desktop\24电工杯\附件1: 各园区典型日负荷数据.xlsx');
% 读取风光发电数据
wind_solar_data = readmatrix('C:\Users\86139\Desktop\24电工杯\附件2: 各园区典型日风光发电数据.xlsx');

% 参数设置
SOC_max = 0.9;
SOC_min = 0.1;
charge_efficiency = 0.95;
discharge_efficiency = 0.95;
electricity_price = 1; % 电网电价 (元/千瓦时)
wind_cost_per_kwh = 0.5; % 风力发电成本 (元/千瓦时)
solar_cost_per_kwh = 0.4; % 光伏发电成本 (元/千瓦时)

% 初始化变量

```

```

num_hours = size(load_data, 2);
battery_capacity = 200; % kWh
battery_power = 100; % kW
battery_SOC = zeros(num_hours,2);
battery_SOC(1) = battery_capacity * SOC_min;

% 计算购电量和弃风弃光量
grid_purchase = zeros(num_hours, 1);
wind_solar_waste = zeros(num_hours, 1);

% 计算储能策略
for t = 2:num_hours
    % 计算可充电量和可放电量
    charge_amount = min(battery_capacity * SOC_max - battery_SOC(t-1),
max(0, (wind_solar_data(t,2) - load_data(t,2)) * charge_efficiency));
    discharge_amount = min(battery_SOC(t-1) - battery_capacity *
SOC_min, max(0, (load_data(t,2) - wind_solar_data(t,2)) /
discharge_efficiency));

    % 更新电池SOC
    battery_SOC(t) = battery_SOC(t-1) + charge_amount -
discharge_amount;

    % 更新购电量和弃风弃光量
    grid_purchase(t) = max(0, load_data(t,2) - wind_solar_data(t,2) -
discharge_amount);
    wind_solar_waste(t) = max(0, wind_solar_data(t,2) - load_data(t,2) -
charge_amount);
end

% 计算风光发电成本
wind_generated = wind_solar_data(:,3); % 风力发电量
solar_generated = wind_solar_data(:,4); % 光伏发电量

wind_cost = sum(wind_generated) * wind_cost_per_kwh;
solar_cost = sum(solar_generated) * solar_cost_per_kwh;

% 计算电网购电成本
grid_cost = sum(grid_purchase) * electricity_price;

```

```

% 总供电成本
total_cost = wind_cost + solar_cost + grid_cost;

% 计算单位电量平均供电成本
total_load = sum(load_data(:,2));
avg_cost_per_kwh = total_cost / total_load;

% 输出结果
fprintf('--- With Storage (Including Wind and Solar Costs) ---\n');
fprintf('Total Grid Purchase: %.2f kWh\n', sum(grid_purchase));
fprintf('Total Wind and Solar Waste: %.2f kWh\n', sum(wind_solar_waste));
fprintf('Wind Power Cost: %.2f Yuan\n', wind_cost);
fprintf('Solar Power Cost: %.2f Yuan\n', solar_cost);
fprintf('Total Grid Cost: %.2f Yuan\n', grid_cost);
fprintf('Total Cost: %.2f Yuan\n', total_cost);
fprintf('Average Cost per kWh: %.2f Yuan/kWh\n', avg_cost_per_kwh);

% 可视化结果
time = (1:num_hours)'; % 时间序列

figure;
subplot(3,1,1);
plot(time, grid_purchase, '-b', 'LineWidth', 1.5);
title('Grid Purchase Over Time (With Storage)');
xlabel('Time (hours)');
ylabel('Grid Purchase (kWh)');
grid on;

subplot(3,1,2);
plot(time, wind_solar_waste, '-r', 'LineWidth', 1.5);
title('Wind and Solar Waste Over Time (With Storage)');
xlabel('Time (hours)');
ylabel('Waste (kWh)');
grid on;

subplot(3,1,3);
plot(time, battery_SOC, '-g', 'LineWidth', 1.5);
title('Battery State of Charge (SOC) Over Time');
xlabel('Time (hours)');
ylabel('SOC (kWh)');

```



```

grid on;

% 保存图表
saveas(gcf, 'results_with_storage.png');

Task One1.3

% 清空环境变量
clear;
clc;

% 加载数据 (从附件1和附件2中提取数据)
load_data = readtable('C:\Users\86139\Desktop\24电工杯\附件1: 各园区典型日负荷数据.xlsx');
pv_wind_data = readtable('C:\Users\86139\Desktop\24电工杯\附件2: 各园区典型日风光发电数据.xlsx');

% 提取各园区的负荷数据和风光发电数据
load_A = load_data.load_A;
load_B = load_data.load_B;
load_C = load_data.load_C;
pv_A = pv_wind_data.pv_A;
pv_C = pv_wind_data.pv_C;
wind_B = pv_wind_data.wind_B;
wind_C = pv_wind_data.wind_C;

% 参数设置
C_grid = 1; % 网购电价 (元/kWh)
C_power = 800; % 储能功率单价 (元/kW)
C_energy = 1800; % 储能容量单价 (元/kWh)
C_PV = 0.4; % 光伏发电成本 (元/kWh)
C_Wind = 0.5; % 风力发电成本 (元/kWh)
eta_charge = 0.95; % 充电效率
eta_discharge = 0.95; % 放电效率
SOC_min = 0.1; % SOC最小值
SOC_max = 0.9; % SOC最大值

% 假设的储能系统配置 (50kW/100kWh)
P_B = 50; % 储能功率 (kW)
E_B = 100; % 储能容量 (kWh)

% 时间步数 (假设为24小时, 每小时一个时间步)

```

T = 24;

% 定义决策变量

P\_grid\_A = sdpvar(T,1); % 园区A从主电网购电量

P\_grid\_B = sdpvar(T,1); % 园区B从主电网购电量

P\_grid\_C = sdpvar(T,1); % 园区C从主电网购电量

P\_B\_in\_A = sdpvar(T,1); % 园区A储能充电功率

P\_B\_out\_A = sdpvar(T,1); % 园区A储能放电功率

P\_B\_in\_B = sdpvar(T,1); % 园区B储能充电功率

P\_B\_out\_B = sdpvar(T,1); % 园区B储能放电功率

P\_B\_in\_C = sdpvar(T,1); % 园区C储能充电功率

P\_B\_out\_C = sdpvar(T,1); % 园区C储能放电功率

SOC\_A = sdpvar(T,1); % 园区A储能SOC

SOC\_B = sdpvar(T,1); % 园区B储能SOC

SOC\_C = sdpvar(T,1); % 园区C储能SOC

% 目标函数: 最小化各园区总成本

objective = sum(P\_grid\_A) \* C\_grid + sum(P\_grid\_B) \* C\_grid +  
sum(P\_grid\_C) \* C\_grid + ...

P\_B \* C\_power \* 3 + E\_B \* C\_energy \* 3 + ...

sum(pv\_A) \* C\_PV + sum(pv\_C) \* C\_PV + sum(wind\_B) \*  
C\_Wind + sum(wind\_C) \* C\_Wind;

% 约束条件

constraints = [];

% 初始SOC

constraints = [constraints, SOC\_A(1) == E\_B \* 0.5, SOC\_B(1) == E\_B \* 0.5,  
SOC\_C(1) == E\_B \* 0.5];

for t = 1:T

% 园区A的约束

constraints = [constraints, pv\_A(t) + P\_grid\_A(t) + P\_B\_out\_A(t) -  
P\_B\_in\_A(t) == load\_A(t)];

constraints = [constraints, 0 <= P\_B\_in\_A(t) <= P\_B];

constraints = [constraints, 0 <= P\_B\_out\_A(t) <= P\_B];

constraints = [constraints, SOC\_min \* E\_B <= SOC\_A(t) <= SOC\_max \*  
E\_B];

if t == 1

constraints = [constraints, SOC\_A(t) == E\_B \* 0.5 + eta\_charge \*  
load\_A(t) \* E\_B];

```

P_B_in_A(t) - P_B_out_A(t) / eta_discharge];
else
    constraints = [constraints, SOC_A(t) == SOC_A(t-1) + eta_charge *
P_B_in_A(t) - P_B_out_A(t) / eta_discharge];
end

% 园区B的约束
constraints = [constraints, wind_B(t) + P_grid_B(t) + P_B_out_B(t) -
P_B_in_B(t) == load_B(t)];
constraints = [constraints, 0 <= P_B_in_B(t) <= P_B];
constraints = [constraints, 0 <= P_B_out_B(t) <= P_B];
constraints = [constraints, SOC_min * E_B <= SOC_B(t) <= SOC_max *
E_B];
if t == 1
    constraints = [constraints, SOC_B(t) == E_B * 0.5 + eta_charge *
P_B_in_B(t) - P_B_out_B(t) / eta_discharge];
else
    constraints = [constraints, SOC_B(t) == SOC_B(t-1) + eta_charge *
P_B_in_B(t) - P_B_out_B(t) / eta_discharge];
end

% 园区C的约束
constraints = [constraints, pv_C(t) + wind_C(t) + P_grid_C(t) +
P_B_out_C(t) - P_B_in_C(t) == load_C(t)];
constraints = [constraints, 0 <= P_B_in_C(t) <= P_B];
constraints = [constraints, 0 <= P_B_out_C(t) <= P_B];
constraints = [constraints, SOC_min * E_B <= SOC_C(t) <= SOC_max *
E_B];
if t == 1
    constraints = [constraints, SOC_C(t) == E_B * 0.5 + eta_charge *
P_B_in_C(t) - P_B_out_C(t) / eta_discharge];
else
    constraints = [constraints, SOC_C(t) == SOC_C(t-1) + eta_charge *
P_B_in_C(t) - P_B_out_C(t) / eta_discharge];
end
end

% 使用linprog进行优化求解
options = sdpsettings('solver', 'linprog', 'verbose', 1);
sol = optimize(constraints, objective, options);

```

```

if sol.problem == 0
    % 提取优化结果
    P_grid_A_opt = value(P_grid_A);
    P_grid_B_opt = value(P_grid_B);
    P_grid_C_opt = value(P_grid_C);
    P_B_in_A_opt = value(P_B_in_A);
    P_B_out_A_opt = value(P_B_out_A);
    P_B_in_B_opt = value(P_B_in_B);
    P_B_out_B_opt = value(P_B_out_B);
    P_B_in_C_opt = value(P_B_in_C);
    P_B_out_C_opt = value(P_B_out_C);
    SOC_A_opt = value(SOC_A);
    SOC_B_opt = value(SOC_B);
    SOC_C_opt = value(SOC_C);

    % 计算总成本和平均供电成本
    total_cost_A = sum(P_grid_A_opt) * C_grid + P_B * C_power + E_B *
C_energy + sum(pv_A) * C_PV;
    total_cost_B = sum(P_grid_B_opt) * C_grid + P_B * C_power + E_B *
C_energy + sum(wind_B) * C_Wind;
    total_cost_C = sum(P_grid_C_opt) * C_grid + P_B * C_power + E_B *
C_energy + sum(pv_C) * C_PV + sum(wind_C) * C_Wind;
    average_cost_A = total_cost_A / sum(load_A);
    average_cost_B = total_cost_B / sum(load_B);
    average_cost_C = total_cost_C / sum(load_C);

    % 显示各园区总成本和平均供电成本
    disp(['园区A总成本', num2str(total_cost_A), ' 元']);
    disp(['园区A平均供电成本', num2str(average_cost_A), ' 元/kWh']);
    disp(['园区B总成本', num2str(total_cost_B), ' 元']);
    disp(['园区B平均供电成本', num2str(average_cost_B), ' 元/kWh']);
    disp(['园区C总成本', num2str(total_cost_C), ' 元']);
    disp(['园区C平均供电成本', num2str(average_cost_C), ' 元/kWh']);

    % 可视化结果
    figure;
    subplot(4,1,1);
    plot(1:T, P_grid_A_opt, '-o');
    title('园区A从主电网购电量');

```

```

xlabel('时间 (h)');
ylabel('功率 (kW)');

subplot(4,1,2);
plot(1:T, P_B_in_A_opt, '-o');
hold on;
plot(1:T, P_B_out_A_opt, '-o');
title('园区A储能充放电功率');
xlabel('时间 (h)');
ylabel('功率 (kW)');
legend('充电功率','放电功率');

subplot(4,1,3);
plot(1:T, SOC_A_opt, '-o');
title('园区A储能SOC');
xlabel('时间 (h)');
ylabel('SOC (kWh)');

subplot(4,1,4);
plot(1:T, load_A, '-o', 1:T, pv_A, '-*', 1:T, P_grid_A_opt, '-x');
title('园区A总负荷与发电量及购电量');
xlabel('时间 (h)');
ylabel('功率 (kW)');
legend('总负荷','总发电量','从主电网购电量');

figure;
subplot(4,1,1);
plot(1:T, P_grid_B_opt, '-o');
title('园区B从主电网购电量');
xlabel('时间 (h)');
ylabel('功率 (kW)');

subplot(4,1,2);
plot(1:T, P_B_in_B_opt, '-o');
hold on;
plot(1:T, P_B_out_B_opt, '-o');
title('园区B储能充放电功率');
xlabel('时间 (h)');
ylabel('功率 (kW)');
legend('充电功率','放电功率');

```

```

subplot(4,1,3);
plot(1:T, SOC_B_opt, '-o');
title('园区B储能SOC');
xlabel('时间 (h)');
ylabel('SOC (kWh)');

subplot(4,1,4);
plot(1:T, load_B, '-o', 1:T, wind_B, '-*', 1:T, P_grid_B_opt, '-x');
title('园区B总负荷与发电量及购电量');
xlabel('时间 (h)');
ylabel('功率 (kW)');
legend('总负荷', '总发电量', '从主电网购电量');

figure;
subplot(4,1,1);
plot(1:T, P_grid_C_opt, '-o');
title('园区C从主电网购电量');
xlabel('时间 (h)');
ylabel('功率 (kW)');

subplot(4,1,2);
plot(1:T, P_B_in_C_opt, '-o');
hold on;
plot(1:T, P_B_out_C_opt, '-o');
title('园区C储能充放电功率');
xlabel('时间 (h)');
ylabel('功率 (kW)');
legend('充电功率', '放电功率');

subplot(4,1,3);
plot(1:T, SOC_C_opt, '-o');
title('园区C储能SOC');
xlabel('时间 (h)');
ylabel('SOC (kWh)');

subplot(4,1,4);
plot(1:T, load_C, '-o', 1:T, pv_C + wind_C, '-*', 1:T, P_grid_C_opt, '-x');
title('园区C总负荷与发电量及购电量');
xlabel('时间 (h)');

```

```

        ylabel('功率 (kW)');
        legend('总负荷','总发电量','从主电网购电量');
    else
        disp('优化问题求解失败');
    end
    % 定义储能配置组合
    P_B_values = [10, 20, 30, 40, 50]; % 储能功率 (kW)
    E_B_values = [50, 100, 150, 200, 250]; % 储能容量 (kWh)

    best_total_cost = inf;
    best_P_B = 0;
    best_E_B = 0;

    for i = 1:length(P_B_values)
        for j = 1:length(E_B_values)
            P_B = P_B_values(i);
            E_B = E_B_values(j);

            % 重新定义决策变量和约束条件
            P_grid_A = sdpvar(T,1); % 园区A从主电网购电量
            P_grid_B = sdpvar(T,1); % 园区B从主电网购电量
            P_grid_C = sdpvar(T,1); % 园区C从主电网购电量
            P_B_in_A = sdpvar(T,1); % 园区A储能充电功率
            P_B_out_A = sdpvar(T,1); % 园区A储能放电功率
            P_B_in_B = sdpvar(T,1); % 园区B储能充电功率
            P_B_out_B = sdpvar(T,1); % 园区B储能放电功率
            P_B_in_C = sdpvar(T,1); % 园区C储能充电功率
            P_B_out_C = sdpvar(T,1); % 园区C储能放电功率
            SOC_A = sdpvar(T,1); % 园区A储能SOC
            SOC_B = sdpvar(T,1); % 园区B储能SOC
            SOC_C = sdpvar(T,1); % 园区C储能SOC

            % 目标函数: 最小化各园区总成本
            objective = sum(P_grid_A) * C_grid + sum(P_grid_B) * C_grid +
                sum(P_grid_C) * C_grid + ...
                P_B * C_power * 3 + E_B * C_energy * 3 + ...
                sum(pv_A) * C_PV + sum(pv_C) * C_PV + sum(wind_B)
                * C_Wind + sum(wind_C) * C_Wind;

            % 约束条件

```

```

constraints = [];

% 初始SOC
constraints = [constraints, SOC_A(1) == E_B * 0.5, SOC_B(1) == E_B
* 0.5, SOC_C(1) == E_B * 0.5];

for t = 1:T
    % 园区A的约束
    constraints = [constraints, pv_A(t) + P_grid_A(t) + P_B_out_A(t)
- P_B_in_A(t) == load_A(t)];
    constraints = [constraints, 0 <= P_B_in_A(t) <= P_B];
    constraints = [constraints, 0 <= P_B_out_A(t) <= P_B];
    constraints = [constraints, SOC_min * E_B <= SOC_A(t) <=
SOC_max * E_B];
    if t == 1
        constraints = [constraints, SOC_A(t) == E_B * 0.5 +
eta_charge * P_B_in_A(t) - P_B_out_A(t) / eta_discharge];
    else
        constraints = [constraints, SOC_A(t) == SOC_A(t-1) +
eta_charge * P_B_in_A(t) - P_B_out_A(t) / eta_discharge];
    end

    % 园区B的约束
    constraints = [constraints, wind_B(t) + P_grid_B(t) +
P_B_out_B(t) - P_B_in_B(t) == load_B(t)];
    constraints = [constraints, 0 <= P_B_in_B(t) <= P_B];
    constraints = [constraints, 0 <= P_B_out_B(t) <= P_B];
    constraints = [constraints, SOC_min * E_B <= SOC_B(t) <=
SOC_max * E_B];
    if t == 1
        constraints = [constraints, SOC_B(t) == E_B * 0.5 +
eta_charge * P_B_in_B(t) - P_B_out_B(t) / eta_discharge];
    else
        constraints = [constraints, SOC_B(t) == SOC_B(t-1) +
eta_charge * P_B_in_B(t) - P_B_out_B(t) / eta_discharge];
    end

    % 园区C的约束
    constraints = [constraints, pv_C(t) + wind_C(t) + P_grid_C(t) +
P_B_out_C(t) - P_B_in_C(t) == load_C(t)];

```



```

        constraints = [constraints, 0 <= P_B_in_C(t) <= P_B];
        constraints = [constraints, 0 <= P_B_out_C(t) <= P_B];
        constraints = [constraints, SOC_min * E_B <= SOC_C(t) <=
SOC_max * E_B];
        if t == 1
            constraints = [constraints, SOC_C(t) == E_B * 0.5 +
eta_charge * P_B_in_C(t) - P_B_out_C(t) / eta_discharge];
        else
            constraints = [constraints, SOC_C(t) == SOC_C(t-1) +
eta_charge * P_B_in_C(t) - P_B_out_C(t) / eta_discharge];
        end
    end

% 使用linprog进行优化求解
options = sdpsettings('solver', 'linprog', 'verbose', 0);
sol = optimize(constraints, objective, options);

if sol.problem == 0
    % 提取优化结果
    total_cost = value(objective);
    if total_cost < best_total_cost
        best_total_cost = total_cost;
        best_P_B = P_B;
        best_E_B = E_B;
    end
end
end
end

% 显示最优储能配置方案
disp(['最优储能功率: ', num2str(best_P_B), ' kW']);
disp(['最优储能容量: ', num2str(best_E_B), ' kWh']);
disp(['最优方案总成本: ', num2str(best_total_cost), ' 元']);

```

## Task Two 2.1

```

% 读取负荷和发电数据
load_data = xlsread('load_data.xlsx'); % 负荷数据
gen_data = xlsread('联合园区供电.xlsx'); % 风光发电数据

```

```

% 数据时间间隔 (小时)
time_interval = 1;
time = 0:time_interval:(size(load_data, 1) - 1) * time_interval;

% 光伏和风电的发电成本 (元/kWh)
cost_pv = 0.4;
cost_wind = 0.5;

% 主电网购电成本 (元/kWh)
cost_grid = 1;

% 初始化变量
total_load = sum(load_data);
total_gen_pv = sum(gen_data(:,1));
total_gen_wind = sum(gen_data(:,2));
total_gen = total_gen_pv + total_gen_wind;
excess_gen = max(0, total_gen - total_load); % 弃风弃光电量
grid_purchase = max(0, total_load - total_gen); % 从主电网购电量

% 计算总供电成本
total_cost_pv = total_gen_pv * cost_pv;
total_cost_wind = total_gen_wind * cost_wind;
total_cost_grid = grid_purchase * cost_grid;
total_cost = total_cost_pv + total_cost_wind + total_cost_grid;

% 计算单位电量平均供电成本
average_cost = total_cost / total_load;

% 显示结果
fprintf('总购电量: %.2f kWh\n', grid_purchase);
fprintf('总弃风弃光电量: %.2f kWh\n', excess_gen);
fprintf('总供电成本: %.2f 元\n', total_cost);
fprintf('单位电量平均供电成本: %.2f 元/kWh\n', average_cost);

% 可视化
figure;

% 负荷与发电量对比
subplot(3, 1, 1);

```

```

plot(time, load_data, 'b-', 'LineWidth', 1.5); hold on;
plot(time, gen_data(:,1), 'r--', 'LineWidth', 1.5);
plot(time, gen_data(:,2), 'g--', 'LineWidth', 1.5);
xlabel('时间 (小时)');
ylabel('功率 (kW)');
title('负荷与发电量对比');
legend('负荷', '光伏发电', '风电发电');
grid on;

% 购电量和弃电量
subplot(3, 1, 2);
bar([grid_purchase, excess_gen], 0.4);
set(gca, 'XTickLabel', {'购电量', '弃风弃光电量'});
ylabel('电量 (kWh)');
title('购电量和弃电量');
grid on;

% 总供电成本
subplot(3, 1, 3);
bar([total_cost_pv, total_cost_wind, total_cost_grid, total_cost], 0.4);
set(gca, 'XTickLabel', {'光伏成本', '风电成本', '购电成本', '总成本'});
ylabel('成本 (元)');
title('总供电成本');
grid on;

```

## Task Two 2.2

```

% 读取Excel文件中的数据
load_data = readmatrix('load_data.xlsx');
wind_solar_data = readmatrix('wind_solar_data.xlsx');

% 提取各园区的负荷数据
load_A = load_data(:, 1);
load_B = load_data(:, 2);
load_C = load_data(:, 3);

% 提取光伏和风电数据
P_pv_A = wind_solar_data(:, 1);
P_pv_C = wind_solar_data(:, 2);
P_w_B = wind_solar_data(:, 3);
P_w_C = wind_solar_data(:, 4);

```

```

% 设定电网购电成本
C_wind = 0.5; % 风电购电成本 元/kWh
C_pv = 0.4; % 光伏购电成本 元/kWh
C_grid = 1; % 元/kWh

% 计算每小时的购电量和弃风弃光电量
E_grid = zeros(24, 1);
E_excess = zeros(24, 1);

for t = 1:24
    % 各园区总负荷
    L_total = load_A(t) + load_B(t) + load_C(t);

    % 各园区总发电量
    P_pv_total = P_pv_A(t) + P_pv_C(t);
    P_w_total = P_w_B(t) + P_w_C(t);

    % 计算净负荷
    L_net = L_total - (P_pv_total + P_w_total);

    if L_net > 0
        % 需要从电网购电
        E_grid(t) = L_net;
    else
        % 弃风弃光
        E_excess(t) = -L_net;
    end
end

% 计算总购电量和总弃风弃光电量
total_E_grid = sum(E_grid);
total_E_excess = sum(E_excess);

% 计算总供电成本
total_cost = total_E_grid * C_grid;

% 计算单位电量平均供电成本
total_load = sum(load_A) + sum(load_B) + sum(load_C);
avg_cost = total_cost / total_load;

```

```

% 设定储能参数
P_bat_max = 10; % kW
E_bat_capacity = 50; % kWh
SOC_min = 0.1;
SOC_max = 0.9;
eta = 0.95; % 充放电效率

% 初始化储能状态
E_bat = E_bat_capacity * 0.5; % 初始状态设为50% SOC

% 计算配置储能后的购电量
E_grid_storage = zeros(24, 1);

for t = 1:24
    % 各园区总负荷
    L_total = load_A(t) + load_B(t) + load_C(t);

    % 各园区总发电量
    P_pv_total = P_pv_A(t) + P_pv_C(t);
    P_w_total = P_w_B(t) + P_w_C(t);

    % 计算净负荷
    L_net = L_total - (P_pv_total + P_w_total);

    if L_net > 0
        % 负荷大于发电，放电
        P_discharge = min([P_bat_max, L_net / eta, (E_bat - E_bat_capacity * SOC_min)]);
        E_bat = E_bat - P_discharge / eta;
        E_grid_storage(t) = L_net - P_discharge;
    else
        % 发电大于负荷，充电
        P_charge = min([-L_net * eta, P_bat_max, (E_bat_capacity * SOC_max - E_bat)]);
        E_bat = E_bat + P_charge * eta;
        E_grid_storage(t) = 0;
    end
end
end

```

```

% 计算总购电量
total_E_grid_storage = sum(E_grid_storage);

% 计算总供电成本
total_cost_storage = total_E_grid_storage * C_grid;

% 计算单位电量平均供电成本
avg_cost_storage = total_cost_storage / total_load;

% 可视化结果
figure;

% 未配置储能情况下的购电量
subplot(3, 1, 1);
plot(1:24, E_grid, '-o', 'DisplayName', '未配置储能');
hold on;
plot(1:24, E_grid_storage, '-x', 'DisplayName', '配置储能');
title('每小时购电量比较');
xlabel('时间 (小时) ');
ylabel('购电量 (kWh) ');
legend('show');

% 弃风弃光量
subplot(3, 1, 2);
plot(1:24, E_excess, '-o', 'DisplayName', '未配置储能');
hold on;
title('每小时弃风弃光量');
xlabel('时间 (小时) ');
ylabel('弃风弃光量 (kWh) ');
legend('show');

% 总供电成本和单位电量平均供电成本
subplot(3, 1, 3);
bar([total_cost, total_cost_storage; avg_cost, avg_cost_storage]);
set(gca, 'xticklabel', {'总供电成本', '单位电量平均供电成本'});
legend({'未配置储能', '配置储能'});
title('总供电成本和单位电量平均供电成本比较');
ylabel('成本 (元) ');

% 显示结果

```

```

fprintf('未配置储能时:\n');
fprintf('总购电量: %.2f kWh\n', total_E_grid);
fprintf('总弃风弃光电量: %.2f kWh\n', total_E_excess);
fprintf('总供电成本: %.2f 元\n', total_cost);
fprintf('单位电量平均供电成本: %.2f 元/kWh\n', avg_cost);

fprintf('配置储能后:\n');
fprintf('总购电量: %.2f kWh\n', total_E_grid_storage);
fprintf('总供电成本: %.2f 元\n', total_cost_storage);
fprintf('单位电量平均供电成本: %.2f 元/kWh\n', avg_cost_storage);

```

### Task Three 3

```

% 读取附件3的数据，保留原始列标题
data = readtable('F:\桌面\jiangongbei\A题\附件3: 12个月各园区典型日风光发电数据.xlsx', 'VariableNamingRule', 'preserve');

% 查看数据的前几行，以确定列标题
disp(data.Properties.VariableNames);

% 提取光伏和风电发电数据
months = 12;
hours = 24;
pv_data = zeros(months, hours);
wind_data = zeros(months, hours);

% 创建列名模式数组
pv_columns = {'光伏出力', '光伏出力_1', '光伏出力_2', '光伏出力_3', '光伏出力_4', ...
              '光伏出力_5', '光伏出力_6', '光伏出力_7', '光伏出力_8', '光伏出力_9', ...
              '光伏出力_10', '光伏出力_11', '光伏出力_12'};
wind_columns = {'风电出力', '风电出力_1', '风电出力_2', '风电出力_3', '风电出力_4', ...
                '风电出力_5', '风电出力_6', '风电出力_7', '风电出力_8', '风电出力_9', ...
                '风电出力_10', '风电出力_11', '风电出力_12'};

for month = 1:months
    % 动态获取列名

```

```

pv_col = pv_columns{month};
wind_col = wind_columns{month};

% 从数据表中提取对应月份的光伏发电数据, 并存储到 pv_data 矩阵中
pv_data(month, :) = data{:, pv_col};

% 从数据表中提取对应月份的风电发电数据, 并存储到 wind_data 矩阵中
wind_data(month, :) = data{:, wind_col};
end

% 检查提取的数据
disp(pv_data);
disp(wind_data);

% 参数设定
PLmax_A = 447 * 1.5;
PLmax_B = 419 * 1.5;
PLmax_C = 506 * 1.5;

% 发电和储能成本参数
Cpv = 2500;
Cw = 3000;
CsP = 800;
CsE = 1800;

% 储能系统SOC范围和效率
SOC_min = 0.1;
SOC_max = 0.9;
eta = 0.95;

% 投资回报期
T = 5;

% 分时电价
peak_price = 1;
offpeak_price = 0.4;

% 负荷增长后的最大负荷
PLmax = [PLmax_A, PLmax_B, PLmax_C];

```



```

% 独立运营：计算每个园区的最优储能配置方案
Ppv = [750, 0, 600]; % A, B, C的光伏装机容量
Pw = [0, 1000, 500]; % A, B, C的风电装机容量

% 假设初始储能配置
Ps = [50, 50, 50]; % kW
Es = [100, 100, 100]; % kWh

% 计算投资成本
cost_pv = Ppv * Cpv;
cost_wind = Pw * Cw;
cost_storage = Ps * CsP + Es * CsE;
total_cost = cost_pv + cost_wind + cost_storage;

% 打印各园区独立运营配置方案及其投资成本
fprintf('园区A投资成本: %.2f元\n', total_cost(1));
fprintf('园区B投资成本: %.2f元\n', total_cost(2));
fprintf('园区C投资成本: %.2f元\n', total_cost(3));

% 联合运营：合并三个园区的负荷和发电量
total_PLmax = sum(PLmax);
total_Ppv = sum(Ppv);
total_Pw = sum(Pw);
total_Ps = sum(Ps);
total_Es = sum(Es);

% 计算联合运营的投资成本
total_cost_union = total_Ppv * Cpv + total_Pw * Cw + total_Ps * CsP +
total_Es * CsE;

% 打印联合运营配置方案及其投资成本
fprintf('联合运营总投资成本: %.2f元\n', total_cost_union);

% 可视化每月光伏和风电发电数据
figure;
for month = 1:months
    subplot(4, 3, month);
    plot(1:hours, pv_data(month, :), 'r', 1:hours, wind_data(month, :), 'b');
    title(sprintf('Month %d', month));

```

```

        legend('PV', 'Wind');
        xlabel('Hour');
        ylabel('Power (kW)');
    end

```

```

% 可视化投资成本对比
figure;
bar([total_cost, total_cost_union]);
title('投资成本对比');
legend('园区A', '园区B', '园区C', '联合运营');
xlabel('园区');
ylabel('投资成本 (元) ');

```

```

% 假设购电成本
purchase_cost = [50000, 60000, 55000]; % 假设的年度购电成本 (元)

```

```

% 可视化购电成本
figure;
bar(purchase_cost);
title('年度购电成本');
xlabel('园区');
ylabel('购电成本 (元) ');

```

```

% 假设回报率
return_rate = [0.15, 0.12, 0.14]; % 假设的回报率 (%)

```

```

% 可视化回报率
figure;
bar(return_rate);
title('投资回报率');
xlabel('园区');
ylabel('回报率 (%) ');

```

```

% 可视化各园区独立运营配置方案
figure;
subplot(2, 1, 1);

```

```

bar([Ppv', Pw']);
title('各园区独立运营配置方案');
xlabel('园区');
ylabel('装机容量 (kW) ');
legend('光伏', '风电');

subplot(2, 1, 2);
bar([Ps', Es']);
title('各园区独立运营储能配置方案');
xlabel('园区');
ylabel('储能容量 (kW / kWh) ');
legend('功率', '能量');

```

```

% 可视化联合运营配置方案
figure;
bar([total_Ppv, total_Pw, total_Ps, total_Es]);
title('联合运营配置方案');
xlabel('配置项');
ylabel('容量 (kW / kWh) ');
xticklabels({'光伏', '风电', '储能功率', '储能能量'});

```